DOCUMENT RESUME

ED 238 406 IR 010 928

AUTHOR Bayman, Piraye
TITLE The Effects of Instructional Procedures on Beginning Programmers' Mental Models.
PUB DATE 29 Aug 83
NOTE 7p.; Paper presented at the Annual Convention of the American Psychological Association (91st, Anaheim, CA, August 29, 1983). For a related document, see IR 010 916.
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Programs; *Computer Science Education; *Concept Formation; Higher Education; *Models; Problem Solving; *Programing; Undergraduate Students
IDENTIFIERS *BASIC Programing Language; Mathematical Aptitude

ABSTRACT

Because previous research by Bayman and Mayer (1983) suggests that individuals who attempt to learn their first computer language develop a variety of misconceptions of the meanings of individual programming statements, this study investigated whether it is possible to enhance beginning programmers' understanding of the programming statements through instructions, and whether individuals' "mental models"--conceptions of what happens inside the computer when a statement is executed--are important for their success in solving programming problems. Subjects were 95 undergraduates unfamiliar with computer programming who studied BASIC using either a standard BASIC manual (Standard Group) or manuals containing procedural and/or pictorial information on the internal workings of the computer (Conceptual Groups). The results showed that individuals with lower mathematics aptitude benefitted from the nonstandard information in their manuals and generated more accurate conceptions of the BASIC statements. Their conceptions of the programming statements seem to have influenced their programming performance as well. Eleven references are listed. (Author/LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

The Effects of Instructional Procedures on Beginning Programmers' Mental Models

Piraye Bayman

University of California, Santa Barbara

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

- This document has been reproduced as received from the person or organization originating it.
  Minor changes have been made to improve reproduction quality.
- Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

ED238406

Abstract

Previous research (Bayman and Mayer, 1983) suggests that individuals who attempt to learn their first computer language develop a variety of misconceptions of the meanings of individual programming statements. This project focuses on two important questions: (1) Is it possible to enhance beginning programmers' understanding of the programming statements through instructions? (2) Are individuals' "mental models"--conceptions of what happens inside the computer when a statement gets executed--important for their success in solving programming problems? Ninety-five undergraduates unfamiliar with computer programming studied BASIC using either a standard BASIC manual (Standard Group) or manuals containing procedural and/or pictorial information on the internal workings of the computer (Conceptual Groups). The results showed that individuals with lower mathematics aptitude benefitted from the nonstandard information in their manuals and generated more accurate conceptions of the BASIC statements. Their conceptions of the programming statements seem to have influenced their programming performance as well.

This paper was presented at the session on "Computer-Related Learning" at the 91st Annual Convention of the American Psychological Association at Anaheim, California, August 29, 1983.

A more detailed report of this project is available from the author at the following address: Piraye Bayman, Department of Psychology, University of California, Santa Barbara, CA 93106.

IR010938

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

Piraye Bayman

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)"

The Effects of Instructional Procedures on Beginning Programmers' Mental Models

Rationale

In an earlier project, Bayman and Mayer (1983) studied computer-naive individuals' understanding of elementary BASIC programming statements. Their findings suggest that beginners might develop a variety of misconceptions of the meanings of individual programming statements in the process of learning their first computer language. The present project attempts to answer the following important questions: (1) Is it possible to foster correct conceptions of the programming statements through instructions? (2) Are individuals with fewer or no misconceptions more successful in solving programming problems that involve the use of elementary programming statements?

Recently, various researchers have argued that learning within complex knowledge domains involves more than the acquisition of new facts, concepts, and rules (Carroll and Thomas, 1982; de Kleer and Brown, 1983; Gentner and Gentner, 1983; Moran, 1981; Norman, 1980; Stevens, Collins, and Goldin, 1978; and Young, 1983). They have shown that individuals develop various mental models in the process of learning the concepts and rules of the new knowledge domain.

In the domain of computers, DuBoulay, O'Shea, and Monk (1981) have explicitly stated that constructs of different programming languages imply different kinds of conceptual models. For example, they argue that an individual learning BASIC learns how to work a BASIC machine and this might be quite different from a LISP machine. Carroll and Thomas (1982) and Moran (1981) suggest that the internal workings of the computer constitute an important part of the user's interface.

Since unguided beginners are most likely to acquire inaccurate mental models (Bayman and Mayer, 1983 ; and Moran, 1981), it is important to find out about instructional procedures that they can benefit from. The main focus of this paper concerns how to influence novices' conceptual models of BASIC in the early phase of learning the language.

Subjects and Design

Ninety-five undergraduates, naive in computers and computer programming, were randomly distributed into five groups: (1) Standard Group, where the subjects study the BASIC statements from a standard BASIC manual, (2) Summary Group, where the subjects study the statements from a manual that states the major actions that take place inside the computer when each statement is executed, (3) Transaction Group, where the subjects study the statements from a manual that states in detail the actions that take place inside the computer for each statement, (4) Diagram Group, where the subjects study the statements from a manual that describes the actions of the computer with pictures for each statement, and (5) Transaction-Diagram Group, where the subjects study the statements from a manual that states in detail the actions that take place inside the computer as well as provides pictures depicting the internal workings of the computer for each statement.

Procedure

Subjects in each group studied the programming statements from their respective manuals at their own pace in two separate one-hour sessions over a 1-2 week period. Following the study sessions, they came to the lab for two separate one-hour sessions to take the following tests: (1) Problem Solving Test, where they were asked to interpret, generate, or debug programs or single programming statements (this test consisted of 18 problems); (2) Fact Retention Test, where they were asked to fill in the blanks of 20 sentences that were directly taken from their manuals; (3) Procedure Specification Test (PST), where they were instructed to write, in plain English, the actions the computer carries out to execute ten different programming statements; and (4) Content Specification Test (CST),

where they were given a hypothetical model of the computer in diagram form and were instructed to fill in the necessary parts of the diagram for the same ten statements included in the PST.

Results and Discussion

Scoring. Each subject's answer for each one of the problems in the Problem Solving Test and each one of the questions in the Fact Retention Test was scored as either correct or incorrect based on the predetermined answers. Slight modifications of the answers were considered acceptable and scored as correct as long as there was no incorrect element in them. For example, the following program for one of the program generation problems was scored as correct although the second and fourth statements are unnecessary:

```
10 INPUT A,B
20 PRINT A,B
30 LET C=A+B
40 PRINT C
50 LET D=C/2
60 PRINT D
70 IF D>50 THEN GO TO 10
80 IF D=50 THEN GO TO 10
90 END
```

Each subject's answer for each one of the programming statements in the Procedure and Content Specification Tests was compared against the relevant set of transactions determined prior to the experiment. The transactions for each statement included in the tests were based on Mayer's (1979) analysis of the elementary BASIC statements. The set of transactions for each statement represents the internal workings of the computer one step at a time during the execution of that statement. For example, the set of transactions for one of the statements included in the tests, 70 LET A=B+1, are the following: (1) Find the number in variable A. (2) Erase that number. (3) Find the number in variable B. (4) Add one to that number. (5) Assign the value of B plus 1 to variable A. (6) Go on to the next statement in the program. Note that transaction #5, Assign the value of B plus 1 to variable A, represents the most crucial action of the computer during the execution of this statement. Hence this transaction is determined to be the key transaction for this particular LET statement. Each subject's answer was scored as correct only if the subject included the relevant key transaction in the answer. An answer such as, "Store the equation A=B+1 in memory." would naturally be scored as incorrect.

The results of this project can best be summarized in terms of the following questions:

(1) Did different instructional procedures influence learners' conceptions of the BASIC programming statements? In an earlier study, Bayman and Mayer (in press) described various conceptions of BASIC statements that novices are most likely to acquire within a typical learning environment. The major concern of the present work is to find out whether learners in the Conceptual Groups--Summary, Transaction, Diagram, and Transaction-Diagram Groups--benefit from the information provided for them in their manuals and develop more accurate conceptions of the BASIC statements as compared to those in the Standard Group. For this purpose, the total number of statements scored as correct was counted for each subject separately for the Procedure and Content Specification Tests. Table 1 shows the proportion of correct conceptions from the two tests for higher and lower aptitude subjects in the five instructional groups. Note that it is customary to look at the effects of the instructional treatments for different aptitude levels (Cronbach and Snow, 1977). This data clearly shows that lower aptitude subjects--those who scored 520 or below in the SAT-Quantitative--in the Conceptual Groups outperform those in the Standard Group in terms of the number of correct conceptions they have acquired. To see if this difference is reliable data for the Conceptual Groups is collapsed and compared against the Standard Group in two 2(Group) by 2(Aptitude) analyses of variance, one

for each test. The results revealed overall significant differences between the groups for both tests [$F(1,89)=5.40$, $p<.02$ for PST, $F(1,89)=5.30$, $p<.02$ for CST]. In addition, there was a significant Aptitude by Group Interaction when data from PST was used [$F(1,89)=5.15$, $p<.03$]. These results suggest that the instructional procedures for the Conceptual Groups influenced only the lower aptitude subjects' conceptions of the BASIC programming statements. Higher aptitude subjects seem to be able to abstract the correct meaning of the statements irrespective of the instructional procedures used in this study.

(2) Did instructional procedures influence learners' ability in solving programming problems? Table 2 shows the proportion of correct answers from the Problem Solving Test for higher and lower aptitude subjects in the five instructional groups. When data from the Conceptual Groups were collapsed and compared against the Standard Group with Group and Aptitude as the between-subjects factors, an Aptitude by Group Interaction effect was observed [$F(1,90)=3.84$, $p<.05$]. This result suggests that instructional procedures provided for the Conceptual Groups enhanced lower aptitude subjects' ability to solve simple programming problems but hindered the ability of the higher aptitude subjects. One possible explanation for this detrimental effect could be that the higher aptitude subjects could not use their processing skills as effectively in the presence of the instructional procedures used in this project.

(3) Are individuals' conceptions of the BASIC statements instrumental for their programming performance? To answer this question, subjects who fall in the first quartile based on the proportion of correct conceptions in the PST were considered as having good mental models. Similarly, those who fall in the last quartile were considered as having poor mental models of the BASIC statements. Subjects classified as having good mental models got at least eight of the ten statements correct. Whereas, subjects classified as having poor mental models got four or fewer of the ten statements correct. A one way analysis of variance was performed on subjects' problem solving test scores with success in having accurate mental models as the between-subjects factor. This analysis revealed that those with good mental models performed significantly better than those with poor mental models [$F(1,42)=49.35$, $p<.0001$]. Hence, having accurate conceptions of the programming statements seems to be instrumental in one's success in using those statements in various programming tasks.

Conclusion

The present study provides evidence that subjects with lower mathematics aptitude benefit from studying descriptions of the internal workings of the computer as they attempt to learn their first computer language. The instructional procedures seem to have helped the beginning programmer (1) in the generation of accurate conceptions of the programming statements, and (2) in the acquisition of elementary programming skills.

References

Bayman, P., and Mayer, R. E. Diagnosis of beginning programmers' misconceptions of BASIC programming statements. Communications of the ACM, 1983, 26, 677-679.

Carroll, J. M., and Thomas, J. C. Metaphor and the cognitive representation of computing systems. IEEE Transactions on Systems, Man, Cybernetics, 1982, 12, 107-116.

Cronbach, L. J., and Snow, R. E. Aptitudes and instructional methods: A handbook for research on interactions. New York: Irvington Publishers, 1977.

de Kleer, J., and Brown, J. S. Assumptions and ambiguities in mechanistic mental models. In D. Gentner and A. L. Stevens (Eds.), Mental Models, Hillsdale, N.J.: Lawrence Erlbaum Associates, 1983.

DuBoulay, B., O'Shea, T., and Monk, J. The black box inside the glass box: Presenting computing concepts to novices. Communications of the ACM, 1981, 14, 237-249.

Gentner, D., and Gentner, D. R. Flowing waters or teeming crowds: Mental models of electricity. In D. Gentner and A. L. Stevens (Eds.), Mental Models, Hillsdale, N.J.: Lawrence Erlbaum Associates, 1983.

Mayer, R. E. A psychology of learning BASIC. Communications of the ACM, 1979, 11, 589-593.

Moran, T. P. The command language grammar: A representation for the user interface of interactive computer systems. International Journal of Man-Machine Studies, 1981, 15, 3-50.

Norman, D. A. What goes on in the mind of the learner. New Directions for Teaching and Learning, 1980, 2, 37-49.

Stevens, A., Collins, A., and Goldin, S. E. Misconceptions in students' understanding. International Journal of Man-Machine Studies, 1979, 11, 145-156.

Young, R. M. Surrogates and mappings: Two kinds of conceptual models for interactive devices. In D. Gentner and A. L. Stevens (Eds.), Mental Models, Hillsdale, N.J. : Lawrence Erlbaum Associates, 1983.

Table 1

Proportion of Correct Conceptions from Procedure and Content Specification Test for Higher and Lower Aptitude Subjects in Each Group

	Procedure Spec Test		Content Spec Test	
Instructional Groups	Higher Apt	Lower Apt	Higher Apt	Lower Apt
Standard (n=19)	.67	.30	.61	.36
Summary (n=18)	.69	.43	.68	.49
Transaction (n=19)	.68	.65	.56	.63
Diagram (n=19)	.65	.49	.61	.51
Tran-Diag (n=18)	.66	.56	.74	.63

Note. The latter four groups constitute the Conceptual Groups.

Table 2

Proportion of Correct Problems from Problem Solving Test for Higher and Lower Aptitude Subjects in Each Instructional Group

Instructional Groups	Higher Aptitude	Lower Aptitude
Standard	.74	.48
Summary	.66	.57
Transaction	.63	.64
Diagram	.64	.53
Tran-Diag	.68	.58